REMARKS

In view of the foregoing amendments, claims 1-16 are pending in this application, of which claims 1, 4, 7, 8, 13, 15 and 16 are independent. Claims 1 and 2 are amended for clarity and without new matter; these amendments and supported, for example, by FIG. 3 and FIG. 4 of the immediate application. Claim 3 is amended to correct a typographical error. Claim 4 is amended to add a period to the end of the last line of the claim. Claims 5 and 6 are amended to correctly show dependency to 'the processor' of claims 4 and 5, respectively.

Claims 8-16 are added without new matter. Support for claim 8 can be found in paragraphs 21 and 22. Support for claim 9 can be found in paragraphs 11, 21 and 22 and in FIG. 4. Support for claim 10 can be found in paragraphs 11 and 12. Support for claim 11 can be found in paragraphs 11, 12 and 24 and in FIG. 4. Support for claim 12 can be found in paragraphs 11 and 22 and in FIG. 4. Support for claim 13 can be found in paragraphs 11, 12, 21 and 22 and in FIG. 4. Support for claim 14 can be found in paragraphs 11, 12, 21 and 22 and in FIG. 4. Support for claim 15 can be found in paragraphs 11, 12, 21 and 22 and in FIG. 4. Support for claim 16 can be found in paragraphs 11, 12, 21 and 22 and in FIG. 4.

CLAIM REJECTIONS

Claims 1 and 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,371,684 to Iadonato et al. (hereinafter "Iadonato") in view of U.S. Patent No. 6,598,149 to Clift (hereinafter "Clift"). Respectfully, we disagree.

To help clarify the differences between the immediate application and Iadonato, the Examiner may find the following description helpful. The immediate application discloses register aliasing for data hazard detection logic of a processor that "simplifies the logic associated with ... data hazards so that a virtual register file may map frames of data to a physical register file of equal or larger size ... without corresponding growth of data hazard detect logic." See paragraph 11 of the immediate application. Complexity within the data hazard detection logic is reduced by aliasing multiple physical register IDs to one register ID within the data hazard detection logic. See paragraph 22 and FIG. 4. For example, FIG. 4 illustrates thirty-two register

IDs, RID(32)-RID(63) within the data hazard detection logic that alias to one-hundred-and-twenty-eight general registers, GR(32)-GR(159), of the processor. Thus, in this example, each register ID within the data hazard detection logic is aliased to four general registers, such that the data hazard detection logic need only compare thirty-two register IDs and not one-hundred-and-twenty-eight. Thus, the complexity of the data hazard detection logic is reduced, since fewer comparisons are required.

Iadonato discloses "a semiconductor floorplan layout for integrating a Data Dependency Checker (DDC) circuit and a Tag Assignment Logic (TAL) of a Register Renaming Circuit (RRC) circuit to conserve valuable semiconductor realestate ... to reduce the distance signals must travel between the DDC and TAL, as well as the distance signals must travel between the TAL and RPM." See Iadonato col. 2, lines 38-47. Iadonato does not therefore reduce complexity of the DDC, only the distance traveled by signals reaching the DDC; it also does not address the problem of complexity within the data hazard detection logic as addressed by the immediate application.

Amended claim 1 recites a method for stacked register aliasing in data hazard detection of a processor, including the steps:

a) identifying a first group of registers within a register file of the processor;

b) aliasing the first group of registers to first register identifiers;

c) detecting data hazards, if any, associated with the first register identifiers;

d) identifying a second group of registers within the register file;

e) aliasing the second group of registers to second register identifiers; and

f) detecting data hazards, if any, associated with the second register identifiers, wherein the first and second register identifiers overlap in hazard detect logic across two or more rows of the register file.

These steps are for example illustrated by FIG. 4 and paragraph 22 of the immediate application, wherein "RID(32:63) maps to each set GR(32:63), GR(64:95), GR(96:127), GR(128:159)," illustrating overlap in hazard detect logic across four rows of the register file. More specifically, the first and second register identifiers each overlap in hazard detection logic across two or more rows of the register file. Continuing with the example of FIG. 4, overlap is shown with GR(32) aliasing to RID(32), GR(64) aliasing to RID(32), GR(96) aliasing to RID(32) and GR(128) aliasing to RID(32). Similarly, GR(33) aliases to RID(33), GR(65) aliases to RID(33), GR(97) aliases to RID(33) and GR(129) aliases to RID(33). As required by claim 1, data hazard detection logic is associated with register identifiers, and, in this example, detects data hazards within register identifiers RID(32:63).

The Examiner asserts that Iadonato discloses this overlap in column 5, lines 26-65 and in FIG. 2. Respectfully we disagree. Iadonato does not disclose register identifier overlap. In fact, Iadonato states that "all source register addresses are compared with all previous destination register addresses." See Iadonato col. 5, lines 34-36. Thus, Iadonato teaches a one-to-one comparison of register files within the DDC and does not teach register identifier overlap.

In FIG. 2, Iadonato discloses a logic matrix that implements address comparison. Iadonato further discloses here that data dependency checker 108 has two sets of inputs and that "the first set includes source address signals from IFIFO 101 for all eight instructions of window 102 ... the second set of inputs includes ... destination address signals ... for all eight instructions." See Iadonato col. 6 line 66 through col. 7 line 6. Iadonato does not therefore disclose first and second register identifiers overlapping in hazard detect logic across two or more rows of the register file, as required by claim 1. Iadonato also clearly does not disclose aliasing, as required by steps b) and e) of claim 1.

Clift discloses "a technique for enhancing performance for code transitions of floating point and packed data modes." See the abstract of Clift. Clift does not overcome the shortcomings of Iadonato. For example, Clift does not teach hazard detection logic, overlap or aliasing, as required by claim 1. Therefore, Iadonato in view of Clift also cannot teach claim 1.

Reconsideration of claim 1 is respectfully requested.

Claim 7 recites that, in data hazard detect logic of a processor of the type having a register file and a register ID file providing row-to-row data hazard detection, an improvement is provided wherein the register file ID aliases row-to-row hazard detection of the register file by common data hazard detection logic for two or more rows of the register file. In an example of row to row hazard detection of claim 7, hazards are detected by identifying matching register IDs of registers within the register file. Claim 7 specifically teaches that the register file ID ***aliases*** row-to-row hazard detection of the register file for two or more rows of the register file. As argued above, the register file ID aliases to two or more registers of the register file. When the data hazard detection logic compares a first register ID to a second register ID, it effectively compares two or more rows of the register file (aliased by the first register ID) to two or more rows of the register file (aliased by the second register ID). That is to say, given a register identifier for one row of the register file, data hazard detection logic does not distinguish between any of the aliased rows within the register file when comparing row-to-row.

For example, again consider paragraph 22 of the immediate application, which describes this example: "the register ID file has 32 register identifiers, then each subsequent set of 32 GRs beginning with GR(32) (e.g., GR(32:63), GR(64:95), GR(96:127) and GR(128:159)) alias respectively to the same 32 hazard detect register identifiers RID(32:63), as illustrated in FIG. 4." Paragraph 22 continues with "register IDs now alias to common hazard detect logic for rows GR(32), GR(64), GR(96), for rows GR(33), GR(65), GR(97), and so on, of register file 114." Thus, in this example, it can be seen that register identifier RID(32) alias to registers GR(32), GR(64), GR(96) and GR(128), such that hazard detection logic does not distinguish between registers GR(32), GR(64), GR(96) and GR(128) when comparing row-to-row; hazard detection logic would identify a hazard when comparing register GR(32) to register GR(128), in this example.

As argued above, in FIG. 2, Iadonato discloses a logic matrix that implements address comparison wherein, as previously noted, "all source register addresses are compared with all previous destination register addresses." See Iadonato col. 5, lines 34-36. Thus Iadonato does not teach of reducing hazard detection logic complexity by comparing groups of ***aliased*** rows. As argued above, Clift does not overcome the shortcomings of Iadonato. Therefore, Iadonato in view of Clift cannot teach claim 7.

Reconsideration of claim 7 is respectfully requested.

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Iadonato in view of Clift and further in view of U.S. Patent No.6,757,807 to Dye (hereinafter "Dye"). Respectfully, we disagree.

Dye discloses a processor that "executes display list commands in processor and coprocessor mode and dynamically switches between these two modes." See Dye col. 3, lines 29-32. Dye does not disclose data hazard detection logic within the processor, nor does Dye teach or suggest aliasing of registers. Dye does not overcome the failings of Iadonato and Clift, argued above. Since claim 2 depends from claim 1 and benefits from arguments presented for claim 1, Iadonato in view of Clift and in further view of Dye cannot anticipate claim 2.

Reconsideration of claim 2 is requested.

Claims 4 and 5 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,826,055 to Wang et al. (hereinafter "Wang") in view of U.S. Patent No. 6,598,149 to Clift (hereinafter "Clift"). Respectfully, we disagree.

Claim 4 recites a processor for processing program instructions, including:

a) a register file;

b) an execution unit having an array of pipelines for processing the instructions and for writing bypass data to the register file; and

c) data hazard detect logic for detecting and aliasing data hazard detection for two or more rows of the register file.

In the immediate application, rows of register files are aliased such that the data hazard detection logic applies to aliased register IDs. See paragraphs 11 and 22 of the immediate application. Wang discloses that "data dependency logic, residing in RRC 204, is used for checking instructions for dependencies, and that "in checking for dependencies, the data dependency checking logic looks at the various registers to determine whether one or more previous instructions must be executed before a subsequent instruction may be executed." See Wang col. 6, lines 58-63. Wang does not disclose or suggest, at least, aliasing as required by element c) of claim 4. As argued above, Clift discloses a technique for enhancing performance for code transitions of floating point and packed data modes." See the abstract of Clift. Since neither Wang nor Clift teaches aliasing, they cannot be reasonably combined to render claim 4 obvious under 35 U.S.C. §103.

Reconsideration of claim 4 is requested.

Claim 5 depends from claim 4 and therefore benefits from like arguments; but in addition claim 5 has other features patentable over Wang and Clift. For example, claim 5 recites a register ID file for facilitating data hazard detection associated with rows of the register file, the register ID file having a plurality of register identifiers, the data hazard detect logic aliasing data hazard detection according to mapping of the register identifiers. Again, claim 5 requires that the register ID file facilitates data hazard detection. As shown in FIG. 4 and described in paragraphs 11 and 22 of the immediate application, the register ID file does not have a one to one relationship with the register file of the processor. In fact, each entry within the register ID file is aliased to two or more registers of the register file. Since data hazard detection logic processed the register ID file, and hence the register aliases, complexity of the data hazard detection logic is reduced. Again, since Wang does not teach aliasing, Wang cannot anticipate claim 5.

Reconsideration of claim 5 is requested.

We appreciate the indication of allowable subject matter of claims 3 and 6.

For the reasons discussed above, we believe that claims 1-16 are allowable. Reconsideration and allowance of all claims are now requested.

There are now 16 claims and 7 independent claims. 20 total claims and 3 independent claims were paid for at the initial filing. Accordingly, a fee for $352 should be changed to Deposit Account No. 08-2025. Applicants believe no further fees are due in connection with this amendment. If any additional fee is due, please charge Deposit Account No. 08-2025.

Respectfully submitted,

By: Curt A. Vock

Curtis A. Vock, Reg. No. 38,356
LATHROP & GAGE L.C.
4845 Pearl East Circle, Suite 302
Boulder, CO 80301
Telephone: (720) 931-3011
Facsimile: (720) 931-3001